



Introduction

This section contains details about the internal processing of the DLGP (device logical gateway process) and the DIOP (device I/O process) and about configuring ABB Automation InfoPlus.21 interface related database records.

Interface Record Creation

For a point to be read or written, the steps below illustrate how a point is registered and used for I/O operations.

1. Create a record using either IoGetDef or IoUnsolDef for read operations or IoPutDef for write operations as the definition record. Information for configuring these records is detailed in **Interface Transfer Records** in this section.
2. Complete the fixed area of the CIM-IO GET, UNSOL or PUT record as described in **Interface Transfer Records**. The fixed area of each record contains information common to all tags configured in the repeat area of the record.
3. Specify the number of points to be processed in the repeat area of this record. Each repeat area occurrence corresponds to a separate ABB Automation tag.
4. Complete the information for each point in the repeat area as described in **Interface Transfer Records**.
5. Now the record is ready to be processed. First, the interface must be running on the client and server sides. After the interface is operational, a record can be activated for processing. To activate a record, the record processing field must be set to *ON*. A record activation can be triggered by setting the *IO_ACTIVATE?* field to *YES*. The field value will be reset to *NO* automatically, and the *IO_LAST_UPDATE* should show a current time-stamp. To have the GET or PUT record periodically activated, a *ScheduledActDef* record can be used. This schedule definition record is described in the **InfoPlus.21 User's Manual**.



Interface Transfer Records

NOTE: This section is only for systems which have InfoPlus.21 as part of the CIM-IO client layer.

The transfer records specify the data to be accessed on the InfoPlus.21 interface server and the location to store the result of the operation in the database. The transfer records are created and configured on the client side. There are two types of transfer record definitions for read operations (IoGetDef and IoUnsolDef). There is one transfer record definition for write operations (IoPutDef). For a read or write transfer, a record needs to be created against one of these definition records and configured for proper operation. All transfer records contain the following information:

- Name of the external task that will process the record.
- Name of the group to which the transfer record belongs.
- Field to enable/disable the transfer operation for all points.
- Field to specify how many points are to be transferred.
- Fields to specify the source to store/retrieve the value depending on the transfer type (read/write).
- Fields to specify the individual point names.
- Fields to enable/disable the individual point transfer.
- Fields to specify the individual point processing, data type, engineering unit conversion and deadband.

When a transfer record is activated, an activation request is sent to the external task that is specified in the record. The client external task then sends the request to the DLGP to be routed to the appropriate DIOP for processing.

NOTE: For more details about the CIM-IO transfer records or other CIM-IO records than are covered in this instruction, refer to the ***CIM-IO User's Guide***.

CIM-IO GET Records

The CIM-IO GET records are used to retrieve data from the device and store the values in the database at a regular time interval which is user definable. The read transfer records

specify where to find the values on the InfoPlus.21 interface server and where to store the values in the database. The read transfer record is shown in Figures 3-1 and 3-2.

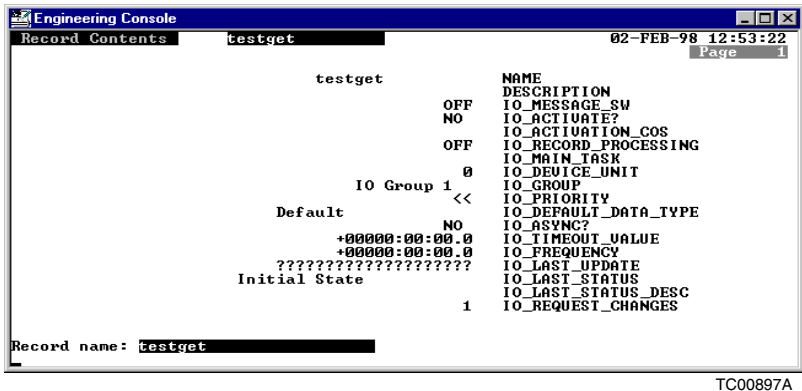


Figure 3-1. Sample ioGetDef Transfer Record Display (1)

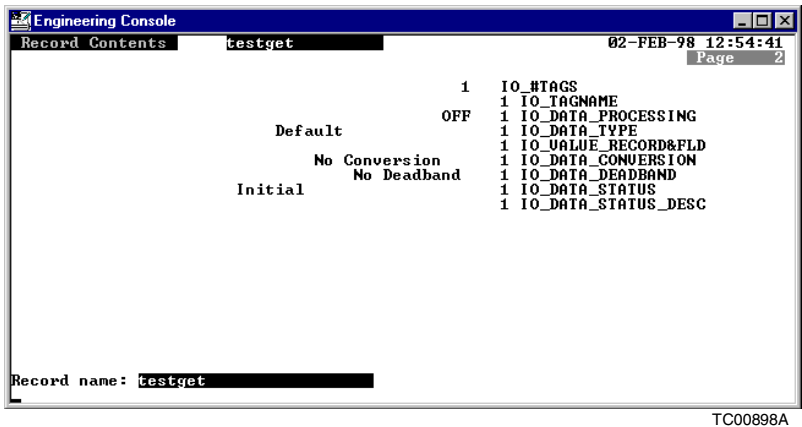


Figure 3-2. Sample ioGetDef Transfer Record Display (2)

Fixed Area Fields

These fields represent values common to all ABB Automation tags within the repeat area of the transfer record. The following fields are in the fixed area of the CIM-IO GET record:

IO_MAIN_TASK - Specifies the client task that will process the record. Set this field to TSK_MPLY to indicate the name of the



CIM-IO client interface task. Now the record can be changed to the usable state.

IO_DEVICE_UNIT - Specifies the unit number. The InfoPlus.21 interface server will not use this field.

IO_GROUP - Specifies the group name to associate with this set of points.

IO_PRIORITY - Specifies the priority CIM-IO should use to send the request to the server when the record is activated. The valid priority range is one to nine. One is the highest priority, and nine is the lowest priority. A value of five to nine should be used for normal read operations.

IO_DEFAULT_DATA_TYPE - Specifies the default data type for all values to be read using this record. If this field value is default, the data type of the target field in the target record (specified by *IO_VALUE_RECORD&FLD*) will be used.

IO_ASYNC? - Specifies whether the GET operations of the polled points are to be processed asynchronously or synchronously. Set this switch to *NO*.

IO_TIMEOUT_VALUE - Specifies the time-out value the client task is to use during read operations. Specify some value such as 00:00:10 for ten seconds. Otherwise, the client task will wait forever for a reply from the server. A negative or null time-out value means the client task will wait forever for a reply from the server.

IO_FREQUENCY - Specifies the scan frequency. The InfoPlus.21 interface server will not use this field.

IO_RECORD_PROCESSING - Enables/disables the processing of all the points defined in the record. Set the field to *ON* for processing of the record.

Read Request Status Fields

The following fields display the status of the read request:

IO_LAST_UPDATE - Displays the time-stamp when the *IO_LAST_STATUS* field was updated.

IO_LAST_STATUS - Displays the overall status of the IoGetDef record.

IO_LAST_STATUS_DESC - Displays the description of the status in the *IO_LAST_STATUS* field.

IO_REQUEST_CHANGES - Fixed field.

IO_#TAGS - Indicates the number of points to be defined in the repeat area. Each point represents one occurrence in the repeat area with fields defined below.

Repeat Area Fields

These fields of the transfer record define the details of the tags to be retrieved from the InfoPlus.21 interface. The following fields are in each occurrence of the repeat area of the CIM-IO GET record. Each point defined is an occurrence within this repeat area with its own set of fields.

IO_TAGNAME - Specifies an address indicating where the data can be found within the InfoPlus.21 interface server. The field is 40 characters in length. Refer to ***Tag Name Addressing*** in **Section 2** for the different address formats.

IO_DATA_PROCESSING - Specifies whether the point will be processed or not. It is an individual switch valid only for that one point. If the field value is *OFF*, the point will not be read from the device, but the rest of the points in the record will be scanned as long as their processing switch is on. If the field value is *ON*, the point value will be read from the InfoPlus.21 interface server.

IO_DATA_TYPE - Specifies the data type of the point in the InfoPlus.21 interface server. The field value supersedes the *IO_DEFAULT_DATA_TYPE* field specified earlier. If the field value is set to default, then the value in *IO_DEFAULT_DATA_TYPE* will be used. If that field is also set to default, then the data type of the database field where the value is to be stored is used.

IO_VALUE_RECORD&FLD - Contains the data record name and field name where the point value retrieved from the interface is to be stored. The data record must already exist.

IO_DATA_CONVERSION - Unused.

IO_DATA_DEADBAND - Unused.



IO_DATA_STATUS - Displays the CIM-IO operation status of the point. The status may indicate a CIM-IO layer error status and/or a server specific error status, that is, an error status from the InfoPlus.21 interface.

IO_DATA_STATUS_DESC - Displays the description of the status in the *IO_DATA_STATUS* field.

CIM-IO Unsolicited Read Transfer Records

The CIM-IO unsolicited records are used to retrieve data from the InfoPlus.21 interface server periodically or whenever the server is ready to send unsolicited data. Unsolicited data is exception-based data that is normally only reported when a value changes in the control system. The unsolicited read transfer records specify where to find the values on the InfoPlus.21 interface server and where to store the values in the database. Using fields in the unsolicited transfer records, the user can make points eligible for unsolicited read processing by declaring them, or removing them from unsolicited read processing by canceling them. The layout of the read transfer record is shown in Figures 3-3 and 3-4.

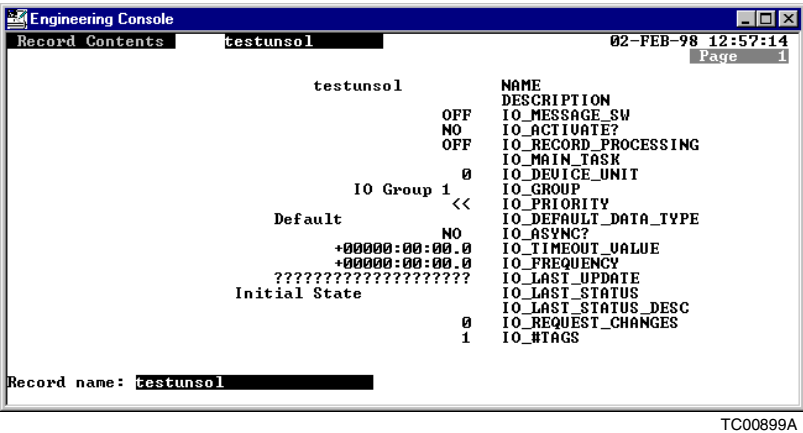
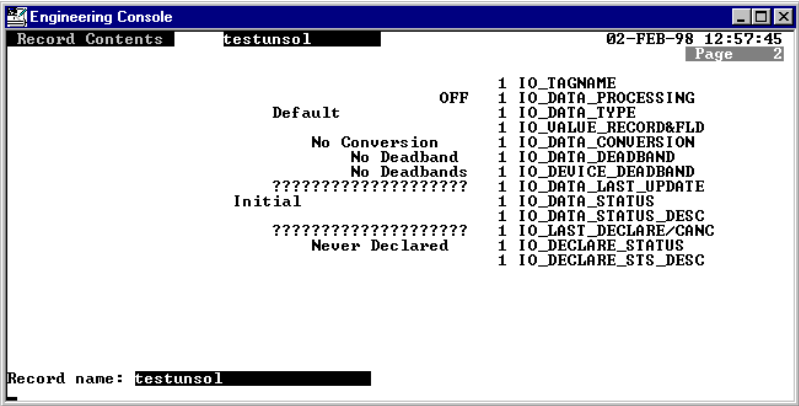


Figure 3-3. Sample IoUnsolDef Transfer Record Display (1)

Fixed Area Fields

These fields represent values that are common to all ABB Automation tags within the repeat area of the transfer record.



TC00900A

Figure 3-4. Sample ioUnsolDef Transfer Record Display (2)

The following fields are in the fixed area of the CIM-IO unsolicited record:

IO_MAIN_TASK - Specifies the client task that will process the record. Set the field to TSK_MBLy for the CIM-IO client interface task. Now the record needs to be made usable.

IO_DEVICE_UNIT - Specifies the unit number. The InfoPlus.21 interface server will not use this field.

IO_GROUP - Specifies the group name to associate with this set of points.

IO_PRIORITY - Specifies the priority CIM-IO should use to send the request to the server when the record is activated. The valid priority range is one to nine. One is the highest priority and nine is the lowest priority. A value of five to nine should be used for normal read operations.

IO_DEFAULT_DATA_TYPE - Specifies the default data type for all values to be read using this record. If this field value is *Default*, the data type of the target field in the target record (specified by *IO_VALUE_RECORD&FLD*) will be used.

IO_ASYNC? - Specifies whether the declaration/cancellation operations of the unsolicited points are to be processed asynchronously or synchronously. Set this switch to *NO*.

IO_TIMEOUT_VALUE - Specifies the time-out value the client task is to use during read operations. Specify some value such



as 00:00:10 for ten seconds. Otherwise, the client task will wait forever for a reply from the server. A negative or null time-out value means the client task will wait forever for a reply from the server.

IO_FREQUENCY - The server sends data to the client whenever it receives data from the InfoPlus.21 interface server that satisfies the deadband criteria. If the ***IO_FREQUENCY*** field value is greater than zero, the server will also send data to the client if it had not sent data to that client for the specified period of time. This ensures the server sends data to the client at some maximum time-out period, overriding the deadband criteria, if any.

IO_RECORD_PROCESSING - Enables/disables the processing of all the points defined in the record. Set the field to *ON* for processing of the record.

Status Fields

The following three fields display the overall status of the unsolicited record:

IO_LAST_UPDATE - Displays the time-stamp when the ***IO_LAST_STATUS*** field was updated.

IO_LAST_STATUS - Displays the overall status of the record including declare/cancel operations and read operations.

IO_LAST_STATUS_DESC - Displays the description of the status in the ***IO_LAST_STATUS*** field.

IO_REQUEST_CHANGES - Fixed field.

IO_#TAGS - Indicates the number of points to be defined in the repeat area. Each point represents one occurrence in the repeat area with fields defined below.

Repeat Area Fields

These fields of the transfer record define the details of the tags to be retrieved from the InfoPlus.21 interface. The following fields are in each occurrence of the repeat area of the CIM-IO unsolicited record. Each point defined is an occurrence within this repeat area with its own set of fields.

IO_TAGNAME - Specifies the address indicating where the data can be found from within the InfoPlus.21 interface server.

The field is 40 characters in length. Refer to **Tag Name Addressing** for the different address formats.

IO_DATA_PROCESSING - Specifies whether the point will be processed or not. It is an individual switch valid only for that one point. If the field value is *OFF*, the point will not be read from the InfoPlus.21 interface server, but the rest of the points in the record will be scanned as long as their processing switch is on. If the field value is *ON*, the point value will be read from the InfoPlus.21 interface server.

IO_DATA_TYPE - Specifies the data type of the point in the InfoPlus.21 interface server. The field value supersedes the **IO_DEFAULT_DATA_TYPE** field specified earlier. If the field value is set to *Default*, then the value in **IO_DEFAULT_DATA_TYPE** will be used. If this field is also set to *Default*, then the data type of the database field where the value is to be stored is used.

IO_VALUE_RECORD&FLD - Contains the data record name and field name where the point value retrieved from the interface is to be stored. The data record must already exist.

IO_DATA_CONVERSION - Unused.

IO_DATA_DEADBAND - Unused.

IO_DEVICE_DEADBAND - Unused.

IO_DATA_LAST_UPDATE - Displays the time-stamp when the **IO_DATA_STATUS** field was updated.

IO_DATA_STATUS - Displays the CIM-IO operation status of the point. The status may indicate a CIM-IO layer error status and/or a server specific error status, that is, an error status from the InfoPlus.21 interface.

IO_DATA_STATUS_DESC - Displays the description of the status in the **IO_DATA_STATUS** field.

IO_LAST_DECLARE/CANC - Unused.

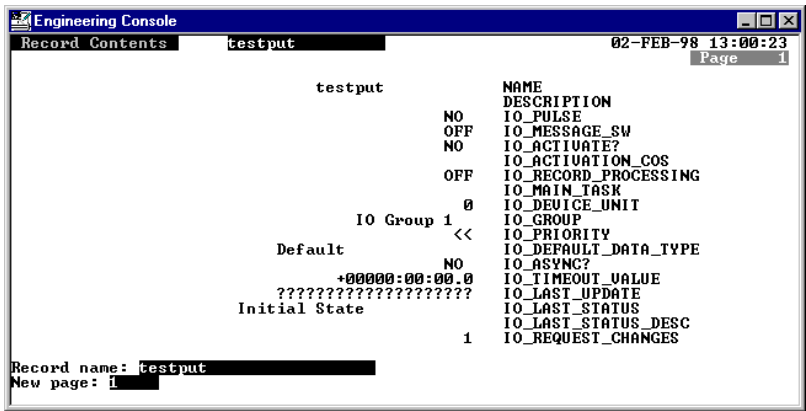
IO_DECLARE_STATUS - Unused.

IO_DECLARE_STS_DESC - Unused.



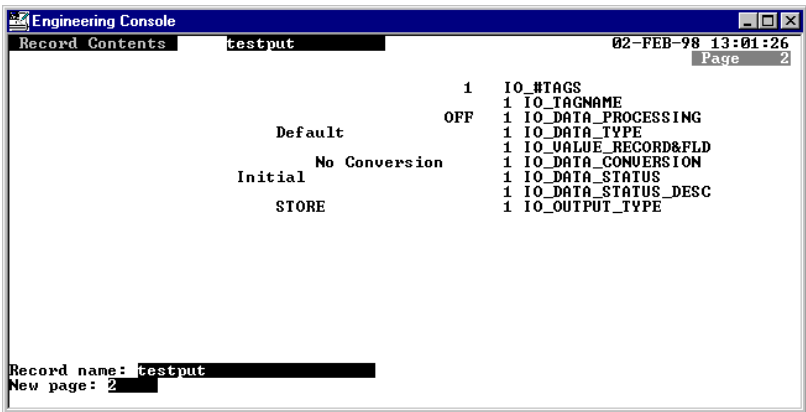
CIM-IO PUT Records

The CIM-IO PUT records are used to write data to the control system. The write transfer records specify where to find the values in the database and where to output the values to in the control system. The layout of the write transfer record is shown in Figures 3-5 and 3-6:



TC00901A

Figure 3-5. Sample IoPutDef Transfer Record Display (1)



TC00902A

Figure 3-6. Sample IoPutDef Transfer Record Display (2)

Fixed Area Fields

These fields represent values common to all ABB Automation tags within the repeat area of the transfer record. The following fields are in the fixed area of the CIM-IO PUT record:

IO_RECORD_PROCESSING - Enables/disables the processing of all the points defined in the record. Set the field to *ON* for processing of the record.

IO_MAIN_TASK - Specifies the client task that will process the record. Set the field to *TSK_MBL* for the CIM-IO client interface task. Now the record needs to be made *USABLE*.

IO_DEVICE_UNIT - Specifies the unit number. The InfoPlus.21 interface server will not use this field.

IO_GROUP - Specifies the group name to associate with this set of points.

IO_PRIORITY - Specifies the priority CIM-IO should use to send the request to the server when the record is activated. The valid priority range is one to nine. One is the highest priority and nine is the lowest priority. A value of two to four should be used for normal write operations.

IO_DEFAULT_DATA_TYPE - Specifies the default data type for all values to be written using this record. If this field value is *Default*, the data type of the target field in the target record, specified by *IO_VALUE_RECORD&FLD*, will be used.

IO_ASYNC? - Specifies whether the PUT points are to be processed asynchronously or synchronously. Set this switch to *NO*.

IO_TIMEOUT_VALUE - Specifies the time-out value the client task is to use during write operations. Specify some value such as *00:00:10* for ten seconds. Otherwise, the client task will wait forever for a reply from the server. A negative or null time-out value means the client task will wait forever for a reply from the server.

Write Request Status Fields

The following fields display the status of the write request:

IO_LAST_UPDATE - Displays the time-stamp when the *IO_LAST_STATUS* field was updated.



IO_LAST_STATUS - Displays the overall status of the IoPutDef record.

IO_LAST_STATUS_DESC - Displays the description of the status in the *IO_LAST_STATUS* field.

IO_REQUEST_CHANGES - Unused.

IO_#TAGS - Indicates the number of points to be defined in the repeat area. Each point represents one occurrence in the repeat area with fields defined below.

Repeat Area Fields

These fields of the transfer record define the details of the tags to be retrieved from the InfoPlus.21 interface. The following fields are in each occurrence of the repeat area of the CIM-IO PUT record. Each point defined is an occurrence within this repeat area with its own set of fields.

IO_TAGNAME - Specifies the address indicating where the data can be found within the InfoPlus.21 interface server. The field is 40 characters in length. Refer to ***Tag Name Addressing*** in **Section 2** for the different address formats.

IO_DATA_PROCESSING - Specifies whether the point will be processed or not. It is an individual switch valid only for that one point. If the field value is *OFF*, the point will not be written to the InfoPlus.21 interface server, but the rest of the points in the record will be scanned as long as their processing switch is on. If the field value is *ON*, the point value will be written to the InfoPlus.21 interface server.

IO_DATA_TYPE - Specifies the data type of the point in the InfoPlus.21 interface server. The field value supersedes the *IO_DEFAULT_DATA_TYPE* field specified earlier. If the field value is set to default, then the value in *IO_DEFAULT_DATA_TYPE* will be used. If that field is also set to *Default*, then the data type of the database field where the value is to be stored is used.

IO_VALUE_RECORD&FLD - Contains the data record name and field name where the point value from the control system is to be written. The data record must already exist.

IO_DATA_CONVERSION - Unused.

IO_DATA_STATUS - Displays the CIM-IO operation status of the point. The status may indicate a CIM-IO layer error status and/or a server specific error status, that is, an error status from the InfoPlus.21 interface.

IO_DATA_STATUS_DESC - Displays the description of the status in the *IO_DATA_STATUS* field.

IO_OUTPUT_TYPE - Unused. Leave at default value *STORE*.

Processing Errors

Communication errors occur when the CIM-IO client tasks encounter problems as they process records. When CIM-IO read and write transfer records are activated, the main CIM-IO client task receives the records for processing. The main CIM-IO client task logs status and error messages about the transfer records as it processes them. This type of error normally results in values being written to the IoLog record. Refer to the ***CIM-IO User's Guide*** for details on the IoLog record. The user can control whether error messages are written to the IoLog record by modifying the *IO_MESSAGE_SW* field in the read and write transfer records. A value of *ON* in the message switch field causes the CIM-IO client tasks to write status and error messages to the IoLog record. The messages going to the IoLog record can be sent to a hard copy printer or to a disk file.

NOTE: The *LOG_DEVICE* field in the IoLog record is formatted by the *REPORT-DEVICES* record. The *LOG_DEVICES* field determines where the error messages are sent. The user can add a new entry to the *REPORT-DEVICES* record and then modify the *LOG_DEVICES* field in the IoLog record to use the new report device. This method allows the user to write the error messages to a printer or a file on disk.

With data stored in it, the IoLog database record looks like:

Record Contents IoLog 05-OCT-1997 15:14:30

IoLog	NAME
CIM-IO Log Disk File	LOG_DEVICE
YES	PRINT_LOG
RTRFURN1AS Success !	LAST_LOG_ENTRY
31-SEP-95 12:43:35.0	LAST_LOG_TIME
83	LOG_SEQUENCE_PRINTED



Record Contents IoLog 05-OCT-1997 15:14:30

83	LOG_SEQUENCE_NUMBER
0	LOGS_ON_DISK
+0:00:00.0	RESCHEDULE_INTERVAL
????????????????????	TIMEOUT_RESCHEDULE
1	# CONDITION FIELDS
	1 CONDITION RECORD
	1 CONDITION FIELD
100	LOGS_IN_MEMORY
RTRFURN1AS Success !	1 LOG_ENTRY
31-SEP-95 12:43:35.0	1 LOG_TIME
WTRFURN3N Some bad points !	2 LOG_ENTRY
31-SEP-95 12:43:30.0	2 LOG_TIME

Internal InfoPlus.21 Interface Server Operation

This section explains the basic internal operation of the InfoPlus.21 interface server. The server consists of two executable programs, each of which is described below.

Device Logical Gateway Process (DLGP)

The DLGP server task is the interface through which all CIM-IO clients communicate with the InfoPlus.21 interface server. When a message is received by the DLGP, it determines the message type and forwards it to the proper destination.

A message can be one of three types:

- **Request** - Messages from CIM-IO clients. CIM-IO request types include GET, GETHIST, PUT, DECLARE, and CANCEL.
- **Reply** - Messages from the DIOP. CIM-IO reply types include GET, GETHIST, PUT, DECLARE, CANCEL, and UNSOLICITED.
- **Notification** - Special messages. CIM-IO notification message types include STOPGET and SHUTDOWN.

Device Input Output Process (DIOP)

The DIOP is the server task that processes all CIM-IO messages from clients and provides the CIM-IO interface with communications to semAPI. Therefore, the DIOP must manage several sets of data including:

- Data from the control system that any CIM-IO clients have requested.
- Information about each CIM-IO client connected to the server.

ABB Automation Point Information

The InfoPlus.21 interface server has been designed with the restriction that all data retrieved from the control system must be retrieved in an unsolicited manner regardless if the client made an unsolicited read or a demand read. Therefore, all values that are read from the control system use the semAPI function `s_read_data_except`. The server stores a buffer with all the point data that clients have requested, and updates this buffer with new control system data exceptions.

The nature of CIM-IO prevents the server from knowing ahead of time which sets of points will be accessed by any clients. Therefore, control system points must be dynamically established whenever the client reads or writes another point. This has three implications for clients to the InfoPlus.21 interface server:

- The client must expect that the first time a point is attempted to be read, it may need to be established in the ABB Automation computer interface. This operation adds time for the initial GET. Time-outs should be adjusted accordingly.
- The client cannot expect that the first time a point is read that the status will be good. If the client has requested a point that has not yet been established, the control system data exception will not be received before a reply is sent. If no other CIM-IO clients have established that point, the server will return a status of `DATA_NOT_RDY` for that point until control system has provided a data exception.



- Establishment of tags occurs online. When a client is establishing a large number of tags, it is possible to internally overwrite some exceptions which are being received while other tags are being established. Once the establish phase is over, normal real-time exception processing is restored.

The core of information stored with each established point in the server's buffer is:

- Loop.
- PCU (process control unit).
- Module.
- Block.
- Point type (ABB Automation point type, indicating which member of the value table to access).
- Value table.
- Time-stamp of last Harmony I/O data exception.
- Client references (list of all clients using this point).

Connected Client Information

The InfoPlus.21 interface server is capable of managing requests from multiple clients. A CIM-IO client is logically connected to the ABB Automation server whenever a request is made to the server, and it is disconnected with a disconnect message or when the client has not used the server for longer than the client time-out.

The core of information stored within each connected client in the server buffer is:

- CIMIO_PROCID (client identification).
- Last access time for client time-outs.
- List of unsolicited lists of tags. Each client will have different lists and may have different deadbands for the same tag. As such, the server manages the deadbands for a tag each time it appears in a list.

SemAPI Calls

The DIOP task makes calls to the following semAPI functions shown in Table 3-1.

Table 3-1. semAPI Functions

Function	Description
s_connect_to_ici	DIOP calls s_connect_to_ici immediately upon starting the DIOP.
s_ici_restart_w	DIOP calls s_ici_restart_w immediately after s_connect_to_ici, and puts the ABB Automation computer interface online. The computer interface cannot be placed offline for point establishing because the server does not know in advance which points all of the clients will be using.
s_establish_import_point_w	Server calls s_establish_import_point_w each time that a new import point is requested for a GET/PUT/DECLARE.
s_establish_export_point_w	Server calls s_establish_export_point_w each time that a new export point is requested for a PUT.
s_read_data_except_w	When the server reads data exceptions from Symphony, the semAPI function s_read_data_except_w is nested in a loop so that it will be repeatedly called until any one of three conditions occurs: <ol style="list-style-type: none"> 1. s_read_data_except_w retrieves less than the maximum exceptions that will fit in the server data area. 2. s_read_data_except_w is called BAILEY_MAX_EXCEPTION_READ_COUNT times. This parameter is configurable on the command line. 3. An exception for any block is received twice during the looping. These termination criteria are necessary to handle the case when ABB Automation exceptions are generated faster than they can be processed.
s_output_control_point_w	DIOP calls s_output_control_point_w each time that PUT is requested to write to an import point.
s_disestablish_point_w	DIOP calls s_disestablish_point_w to disestablish tags for which there are no longer any active clients.
s_disconnect_from_ici	DIOP will call s_disconnect_from_ici upon successfully receiving a disconnect notification from the DLGP.

Time-Stamps

Any CIM-IO replies from the DIOP that return time-stamps use the ABB Automation times that came with the last control system data exception for that block. The control system time is converted to GMT.

